

漢字を正しく書くための学習を支援する 生成AIを用いたツールの開発と評価

鎌田真央¹ 加藤直樹²

概要：本研究では、学習者の書いた手書きの漢字に対する字形を損なう誤りの判定に生成AIを用いた手法を提案し、それを実装した漢字学習支援ツールの開発を行った。さらに生成AIによる指摘と開発したツールの各機能が、学習者の漢字学習を支援するのに有用であるかについて評価実験を行った。評価実験の結果、生成AIによる指摘では特定の問題において指摘の精度や重要性の判断について課題があることが明らかとなった。一方で指摘の内容と誤り箇所の特定に関して、有用であることが示された。また本ツールに実装された各機能は、漢字学習を支援するのに有用であることが示された。

キーワード：漢字書字、書字誤り、漢字指導、生成AI、誤り検出、学習支援、フィードバック

Development and Evaluation Tools Using Generative AI to Support Learning for Writing Kanji Correctly

MAO KAMATA^{†1} NAOKI KATO^{†2}

1. はじめに

1.1 研究背景

GIGAスクール構想によって、小中学校の児童生徒がノート型やスレート型のPCなどの学習者用端末を活用できる環境が整備された[1]。教員は子ども一人一人の実態に応じた適切な教育を行うために、デジタルドリルなど様々なデジタルツールを活用することが求められている。

「GIGAスクール構想のもとでの校務DX化チェックリスト」によると、公立小中学校において、「宿題をクラウドサービスやデジタルドリル教材を用いて実施・採点していますか」という質問に対して、一部していると回答した学校が77.3%、児童生徒一人一人に配備された端末を持ち帰って家庭で利用させていると回答した学校が82.9%となっており[2]、学習者は、デジタルドリル教材を用いることで、学習結果について即時に採点・フィードバックを受け取れるなど、場所や時間を問わず学習に取り組むことが可能となってきた。

デジタルドリルを使用する学習の一つとして漢字学習がある。漢字学習では、漢字の形、読み方や意味、筆順などを学ぶ。正しい漢字の評価について文化庁の常用漢字表の字体・字形に関する指針では、『手書き文字と印刷文字の表し方には習慣の違いがあり、一方だけが正しいのではない、字の細部に違いがあっても、その漢字の骨組みが同じであれば、誤っているとはみなされないとされている。しかし現状の漢字指導では、文字の細部に必要以上の注意

が向けられ、本来であれば問題にならない違いによって、漢字の正誤が決められる傾向が生じており、字体や字形についての周知が不十分である』とされている[3]。しかし、学習者の文字の個人差などが誤りとされる、とめ・はね・はらいなどの字形に関する指導が細かく行われるなど、正しい漢字の評価は曖昧なところがあり、結果として、学習者の学習意欲の低下や、主体的に学ぶ姿勢を妨げることが懸念されている。

既存のデジタル漢字ドリルでは、パターンマッチングや機械学習を用いて画数や筆順、文字の始点や終点などの誤りの指摘が行えるが、突き出しの有無や、曲がり具合、線の長さの偏りなど、字形を大きく損なうような漢字の誤りに対して指摘を行うことが難しいといった課題がある。ここで、大規模言語モデル（Large Language Models）で構成される生成AIは、空間と時間の線形表現を理解することが発見されており[4]、これと関連して筆記した字形を定義する筆跡の座標点列を理解できる可能性がある。座標点列の理解を通して正しい字形について判定することができれば、誤り部分を明確にし、学習者の様々な字形に対応した適切なフィードバックを行うことができる可能性がある。

1.2 研究の目的

本研究では、学習者が書いた様々な字形の漢字に含まれる、字形を大きく損なう誤りの判定・指摘を行うことに生成AIを用いた手法を提案し、その手法を実装した漢字学習ツールを開発・評価することを通して、手法とツール

¹ 東京学芸大学大学院
Graduate School of Education, Tokyo Gakugei University.
² 東京学芸大学
Tokyo Gakugei University

の有用性を検証することを目的とする。

2. 先行研究と製品

2.1 スマイルゼミ

学習者の書いた漢字に対し、どの部分が正しくないかを指摘することのできるツールとしてジャストスマイルが提供するスマイルゼミがある。このツールでは、画数や筆順、文字の始点や終点等の誤りを指摘することができる。しかし始点と終点が正しい位置であれば途中どのように文字を記述しても正しいとされてしまう。また、お手本アニメーションと比較し学習者がどのように文字を記述したかを確認する機能がある。

2.2 iLabo書字学習システム

漢字書字のきれいさを判定できるツールとして、iLabo書字学習システムがある。これは、リアルタイムで画数・筆順を高速・高精度で判定し、多様な項目で文字の「きれいさ」を判定する。また閾値を設定することで判定のレベルを使用者に合わせて調整することができる。しかし、これらの判定は、漢字の字形を大きく損なわない誤りに関する指摘であり、本来問題にならない違いによって正誤が決められている。

2.3 先行研究

古積らは、文字を正しく綺麗に書くために筆順やとめと払い、はねなどの字形を評価する手書き漢字学習支援システムを開発した。このシステムでは、ユーザーの入力パターンと手本文字となる標準パターンのストロークとの差を評価して、画のとめと払い、はね、接続、交差、角度、長さなどの指摘を行える[5]。

しかし、筆順やとめ・はね・はらいといった細部の特徴は、大きく字形を損なう誤りに直結しているとは言えず、さらに字形全体が大きく崩れている場合には、標準パターンと画ごとの対応付け自体が困難となる可能性がある。

3. 手書き漢字の誤りを判定する手法の設計

本章では、学習者が書いた手書き漢字に対して、字形を大きく損なう誤りの判定・指摘に生成AIを用いる手法の設計について述べる。

3.1 基本方針

既存のデジタル漢字ドリルでは、機械学習を用いて画数や筆順、文字の始点や終点などの情報から誤りの指摘が行われている。特に、とめ・はね・はらいといった字形を大きく損なわない細部の特徴は強調して判定される一方で、突き出しの有無や曲がり具合や線の長短などの字形を損なう可能性のある誤りは十分に指摘されていない。

そこで本研究では、字形を大きく損なう可能性のある突き出しの有無や線の長短、数、向きを指摘できるようにすることを目指す。

3.2 正しい漢字と漢字の誤りの定義

本研究において「正しい漢字」とは、堀田らが類型化した誤り[6]の中から、とめ・はね・はらいに関する誤りと異字となる項目を除外した「漢字の字形を大きく損なう可能性のある誤り」がないものとする。常用漢字表の字体・字形に関する指針においても、とめ・はね・はらいの有無は、個人の字形によって異なり、漢字の誤りとされる字形の誤りとは言えないことが示されており、本研究における「正しい漢字」の判定には用いないこととする。

図1に、この「正しい漢字」の字形を損なうことになる「漢字の誤り」を記す。

3.3 判定手法の概要

本研究では、前節で定義した漢字の誤りの判定に生成AIを用いることを試みる。従来の機械学習モデルでは、誤り判定や指摘の精度を向上させるために大量の学習データや特徴量の設計が必要であり、データ収集や前処理に多くの時間と労力を要していた。しかし、大規模言語モデルは事前に学習された汎用的なモデルであるため、漢字の筆跡データや正解となるデータを少量与えるだけで、ストロークの特徴や字形の誤りを判定し、指摘を行うことが期待できる。

学習者が書いた手書き漢字の誤りの判定と指摘に生成AIを用いて行うためには、手書き漢字を表す入力データ、入力データに対して正確な誤りの指摘を行うための比較対象となる正解データ、及びそれらを用いて漢字の誤りを指摘できるプロンプトが必要となる。次節からこれらの設計について述べる。

3.4 入力データ

学習者が書いた手書き漢字の誤りを判定・指摘するためには、その手書き漢字を表すデータ群が必要となる。具体的なデータとしては、画数、筆点列座標、変位、速度、加速度、筆圧、文字認識照会結果、画像を候補として選定した。各データがどのような情報を含んでいるかと必要とした理由について次に示す。

(1) 画数

手書きで書かれた漢字のストローク（漢字を書く際に、ペンや指を画面から離さずに書かれる一つの連続された線）の数を画数として入力データ候補とする。画数は、生成

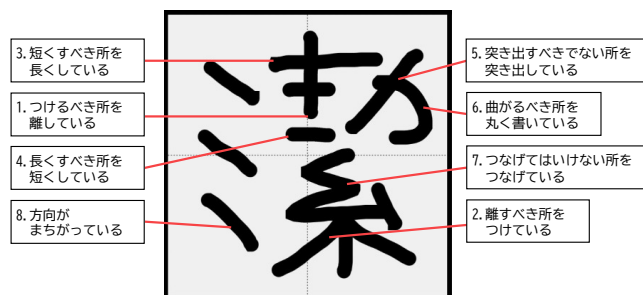


図1 漢字の字形を損なう可能性のある誤り

AIに手書き漢字をストローク単位で正しく認識させるために必要と判断した。

(2) 筆点列座標

手書きで書かれた漢字のストロークを構成する筆点の座標を入力データ候補とする。筆点列座標は、手書きされたストロークを構成する筆点の座標データであり、各ストロークの形状や位置関係を評価するために必要と判断した。なお、各筆点列には何画目のストロークを構成しているかのラベルを付与している。

(3) 変位

筆点列の連続する二点の距離（ベクトルの長さ）を変位として入力データ候補とする。変位は、ストロークの方向や動きの特徴を評価するために必要と判断した。変位 d_n は、 n 番目の点の座標 (x_n, y_n) と $n+1$ 番目の点の座標 (x_{n+1}, y_{n+1}) を用いて、次の計算式で求める。

$$d_n = \sqrt{(x_{n+1} - x_n)^2 + (y_{n+1} - y_n)^2}$$

(4) 速度

漢字学習教材に記述された筆跡の速度を入力データ候補とする。速度は、筆記動作の特徴を捉えるための補助情報として必要と判断した。前述した変位を次式のように、各点が筆記された時間の差分である経過時間 Δt で割ることで、速度 $v_n [px/s]$ を求める。

$$\Delta t = \frac{t_{n+1} - t_n}{1000}, v_n = \frac{d_n}{\Delta t}$$

(5) 加速度

漢字学習教材に記述された筆跡の加速度、つまりペンや指が触れた位置の移動速度の変化率を入力データ候補とする。加速度も、筆記動作の特徴を捉えるための補助情報として必要と判断した。前述した速度を用いて、加速度 $a_n [px/s^2]$ を求める。

$$a_n = \frac{v_{n+1} - v_n}{\Delta t}$$

(6) 筆圧

筆圧としてペンが画面に及ぼす圧力を入力データ候補とする。圧力は 0 から 1 の範囲で値を取り、0 を最低値、1 を最大値とする。

(7) 文字認識照会結果

手書き漢字がどの「漢字」であるかとして、文字認識処理の結果、及びその結果が書くべき漢字であったかの正誤を入力データ候補とする。これにより既存のデジタルドリルでは正解とされてしまうような線の長さや偏りなどの字形の誤りに対しても、正解漢字との対応関係を踏まえた指摘ができるようになることが期待される。

(8) 画像（オフラインデータ）

学習者が画面に書いた漢字の画像（オフラインデータ）を入力データ候補とする。生成AIは画像の内容を認識することが可能である。画像は、座標点列からは十分に把握しきれないストローク同士の間隔やバランスなどの空間的・視覚的特徴を取得できる可能性があるため、漢字全体の構造を把握するための補助情報として必要であると判断した。

3.5 正解データ

学習者が書いた手書きの漢字の誤りを判定・指摘するために、比較対象となる本来書くべきであった漢字を表すデータ（正解データ）が必要となる。このデータの候補として、正解漢字、正解漢字の筆点列座標、画数、正解漢字の画像を選定した。

(1) 正解漢字

学習者が書くべき漢字を正解漢字として正解データ候補とする。このデータは、学習者の書こうとしている漢字を生成AIが正確に把握し、誤りを判定するために必要だと判断した。生成AIが漢字の指摘を行う漢字を特定することにより、学習者へ適切なフィードバックを行うことが期待される。

(2) 正解漢字の筆点列座標

正解漢字を構成する筆点列座標を正解データ候補とする。正解漢字の筆点列を渡すことで、学習者が書いた漢字とストローク単位で比較し誤りを指摘することが期待される。

(3) 画数

正解漢字の画数を正解データ候補とする。学習者が書いた漢字の画数と正解漢字の画数を照らし合わせ、画数が異なる場合に誤りがあるという指摘を行うことが期待される。

(4) 正解漢字の画像

正解漢字の画像を正解データ候補とする。学習者が書いた漢字のストロークと画像を照らし合わせ、画像との違いから字形を大きく損なう誤りについて指摘することが期待される。

3.6 プロンプト

学習者が書いた漢字を表す入力データと正解データから、漢字の誤りを判定・指摘を行うための、生成AIに与える命令であるプロンプトに必要な要素を次に示す。

(1) 生成AIの役割

生成AIがフィードバックを行う役割であることを伝える。これにより生成AIの役割を明確にし、求められる機能や振る舞いを限定し、プロンプト設計の際に、指示の焦点を定めることができることが期待できる。

(2) 目的

生成AIは多様なタスクに対応できる汎用的なモデルであるため、漢字の誤りを効率的かつ正確に特定し、学習者

に必要な情報や指摘を適切に提供させる必要がある。これにより余分な処理を省き、学習者に必要な情報を提供することが期待される。

(3) 入力データの定義

入力データを明確に定義することで、生成AIが学習者の筆跡の特徴を正確に把握し、誤りのある箇所を適切に指摘することが期待される。

入力データとは、学習者が書いた漢字のストロークデータやその漢字の文字認識照会結果などの筆記によって得られるデータであり、生成AIが誤りを判定するための分析対象である。

(4) 正解データの定義

生成AIが学習者の漢字が正しいか判定する基準として正解データを明確に定義する必要がある。これを基に、誤りのある箇所を特定し、どのように修正すべきかをフィードバックすることが期待される。

正解データとは、本ツールが持つ正しい漢字に関する情報であり、生成AIが入力データとの比較を行う基準となる。

(5) 評価条件

生成AIがどの基準に基づいて、漢字の誤りの箇所を判定するかを示すために必要である。評価基準を明確にすることで、生成AIからの出力に一貫性を持たせ、学習者にとってわかりやすいフィードバックを行えるようになることが期待される。

(6) 観点

生成AIが誤りを指摘する際に使用する字形を大きく損なうような誤りの具体例を示すために、必要であると判断した。観点を明確にすることで、生成AIが誤りの特定に際して、曖昧さを排除し、具体的かつ適切な指摘を行うことが期待される。

(7) 出力形式

生成AIが提供するフィードバックの形や表現を指示する必要がある。これにより学習者に伝わりやすい表現や一貫した形で学習者にフィードバックを提供することが期待される。

4. 判定に有効なデータの検証

4.1 評価用ツールの実装

前章で提案した生成AIによる誤り判定手法を検証するために、それを実装したツールを作成した。

本ツールは利用するOSを限定しないために、Webアプリケーションとした。また、本ツールはVisual Studio Codeを用いて開発を行った。開発言語は、JavaScript、HTML、CSSを使用した。実装に当たり、文字の誤りを判定・指摘する生成AIとして、Open AIのGPT-4o APIを採用した。

本ツールの画面構成を図2に示す。画面上の漢字練習用マス(①)に文字を書き込み、採点ボタン(②)を押す

ことで、フィードバックエリア(③)に誤りの指摘が表示される。

4.2 検証方法

提案した生成AIによる誤り指摘手法の検証を行うために、候補として選定した各データについて、全てのデータを揃えた状態からそのデータのみを除いた場合に、生成AIが行うフィードバックにどのような違いが生じるかを調査した。検証に用いた誤りを含んだ入力データは、3.2節で定義した誤りを含むようにツール上に書き込んだ。また、正解データは、標準的な筆順に基づいてお手本漢字をなぞり書きし、その筆跡情報を専用スクリプトで保存することで構築した。

なお、本検証で用いたプロンプトを図3に示す。今回、プロンプトの妥当性は検証の対象としなかった。

4.3 入力データの検証

各入力データ候補が、誤り指摘に与える影響とその有効性を評価した結果を以下に示す。

(1) 画数

画数を入力データとして渡した場合、何画目に誤りがあるかについて指摘を行えた。一方で、画数を渡さない場合は、座標に基づく指摘にとどまり、誤り箇所を画数として明示することはできなかった。

この結果から、画数は有効な入力データであると言える。

(2) 筆点列座標

筆点列座標を入力データとして渡した場合、ストローク同士のつながりに関しての指摘を行えた。一方で、筆点列座標を渡さない場合は、ストロークごとの変位や速度、加速度などの指摘にとどまり、ストローク間の関係性に基づいた指摘は行えなかった。

この結果から、筆点列座標は有効な入力データであると言える。

(3) 変位

変位を入力データとして渡した場合、ストロークの向きやその変化についての指摘を行えた。一方で変位を渡さない場合は、画数や座標に基づく位置的な指摘にとどまり、

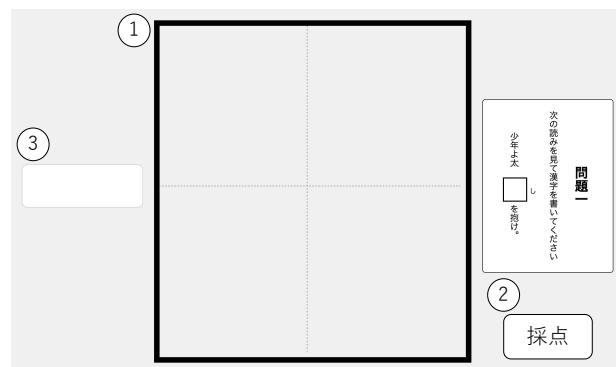


図2 画面構成

あなたは、小学校の国語教師です。以下の指示に沿って、小学生に漢字の誤りをフィードバックしてください。

#目的:漢字の誤りを指摘する。

#条件:

- ・正解の漢字とユーザーが書いた漢字（ストロークデータ）を比較し、ストロークデータの中で正解の漢字の字形を損なう重大な誤りがあれば、誤りのある画のみ指摘してください。
- ・誤りの指摘には明確な根拠を持って、一貫性のある指導をしてください。
- ・漢字の基本的な字形を崩すような重大な誤りが無い判断したら褒めてください。
- ・誤りの種類は観点を参考にすること
- ・各ストロークの詳細データは以下の形式で提供されます:
 - 各ストロークは複数のデータポイントで構成されています。
 - データポイントには、画数（strokeCount）、X座標（x）、Y座標（y）、変位（displacement）、速度（velocity）、加速度（acceleration）、筆圧（pressure）が含まれます。
 - 画数ごとに配列に分け、各画には複数のデータポイントが含まれます。
- ・指摘は簡潔かつ具体的にしてください。（30字以内）。
- ・小学生にもわかりやすい文章にしてください。
- ・特に何画目（数字）が違うかを伝えてください。

#観点:

- ・基本的な漢字の形を重要視し、厳しすぎないように。よくできたときはしっかりと褒めるようにしてください。
- ・部首やつくり
- ・線の長短、数
- ・線の向き（上から下、左から右）
- ・特に以下の点は、文字の字体を大きく損なう可能性がある誤りですので重要視してください。
 1. つけるべき所を離して書いている。
 2. 離すべき所をつけて書いている。
 3. 短くすべき所を長くしている。
 4. 長くすべき所を短くしている。
 5. 突き出すべきでない所を突き出している。
 6. 曲がるべき所を丸く書いている。
 7. つなげてはいけない所をつなげて書いている。
 8. 方向がまちがっている。

#出力形式:箇条書き

図 3 プロンプト

曲がりや丸みに関する具体的な指摘は行えなかった。

この結果から、変位は有効な入力データであると言える。

(4) 速度

速度を入力データとして渡した場合、筆記動作の速さや変化に基づいて、とめ・はね・はらいなどの字形を大きく損なう誤りではない特徴が指摘された。一方で速度を渡さない場合は、主に画数や座標に基づいて誤り箇所の指摘が行われていた。

この結果から、速度を入力データとして生成AIに提供することは誤りの指摘が過剰になる可能性があり、精度が減少することが考えられるため、有効なデータとは言えない。

(5) 加速度

加速度を入力データとして渡した場合、速度を入力データとして渡した際と同様に、加速度の変化に基づいて、とめ・はね・はらいなどの特徴が指摘された。一方で加速度を渡さない場合でも、主に画数や座標に基づく誤り箇所の指摘が行われていた。

この結果から、加速度は有効な入力データとは言えない。

(6) 筆圧

筆圧を入力データとして渡した場合、筆記時の力加減に基づいて、とめ・はね・はらいなどに関する指摘が多く見られた。しかし筆圧の強弱が字形を大きく損なう誤りに直結するとは言いきれず過剰な指摘により学習者に不要な修正を促す可能性が考えられる。一方で、筆圧を渡さない場合でも、主に画数や座標に基づく誤り箇所の指摘が行われていた。

この結果から、筆圧は有効な入力データとは言えない。

(7) 文字認識照会結果

文字認識結果と正解漢字との正誤を渡すことで、文字認識で正解と判断された漢字であっても字形を損なうような誤りを指摘することが可能であった。一方で文字認識結果を渡さなかった場合、誤りの指摘以前に文字の判別自体が曖昧になる傾向が見られた。これは、他の入力データからのみ評価が行われ、場合によっては別の似た漢字が正解と思い込んでの結果を返していることが考えられた。

この結果から、文字認識照会結果は有効な入力データであると言える。

(8) 画像

画像を入力データとして渡した場合、漢字全体の字形や空間的配置に基づく指摘が行われることが期待されたが、実際の出力結果において顕著な差は確認できなかった。また、画像を渡さない場合でも指摘内容に変化は見られなかった。

この結果から、画像は有効な入力データとは言えない。

4.4 正解データの検証

(1) 正解漢字

正解漢字を正解データとして渡した場合、学習者が書いた漢字を基準と照合し、誤りの指摘を行うことが可能であった。一方で正解漢字を渡さない場合は、学習者が書いた漢字の正誤を判断することができない場合が多くみられた。

この結果から、正解漢字は有効な正解データであると言える。

(2) 筆点列座標

正解漢字の筆点列座標を正解データとして渡した場合、学習者の筆跡との差分から突き出しや線の向きについて指摘をすることが可能であった。一方で、筆点列座標を渡さ

なかった場合は、指摘に明確な根拠を欠いた内容が含まれ、内容が曖昧となる傾向がみられた。

この結果から、筆点列座標は有効な正解データであると言える。

(3) 画数

正解漢字の画数を正解データとして渡した場合、画数の誤りについての指摘を行っていた。しかし、続け字や一画を複数画に分けて書いた場合、画数が一致しないために誤りとして過剰に判定される可能性があり、判定精度に限界があることが示唆された。

この結果から、画数は有効な正解データであると言えない。

(4) 正解漢字の画像

画像を正解データとして渡した場合、位置的な差異について言及することがあった。一方で画像を渡さなかった場合でも、正解の筆点列座標を渡した時と同様に突き出しや線の向きなどについて指摘が行われていた。本検証では、生成AIが画像情報をどの程度正確に利用しているかについて明確な根拠は確認できず、画像が誤り指摘において明確な役割を果たしているとは言えなかった。

この結果から、画像は有効な正解データとは言えない。

4.5 考察

入力データとして提供する画数、筆点列座標、変位、文字認識照会結果はそれぞれ明確な役割を果たし、誤り箇所の正確な特定や指摘内容の具体性を向上させる点で非常に有効であることが確認された。一方で速度、加速度、筆圧といったデータは誤りの指摘を過剰にする傾向があり、学習者に不適切な修正を促す可能性があるため生成AIに渡すべきデータとしては適切ではないと考えられる。これらのデータが字形を損なう誤りに直接的に寄与するとは言い難く、誤り指摘の精度を下げる可能性が高い。

次に正解データとして提供すべきデータとして正解漢字、筆点列座標が重要な役割を果たすことが示された。正解漢字を基準とすることで学習者が書いた漢字の誤りを評価することが可能となるが、根拠のない指摘を防ぐためには筆点列座標といった詳細なデータの併用が必要である。

5. 漢字学習AI支援ツールの開発と評価

本章では、学習者の漢字学習を効率化し、より効果的に支援するための漢字学習教材として、3章4章で設計検証した手法を利用した漢字学習AI支援ツールの開発とその評価について述べる。

本ツールはOSを限定しないために、Webアプリケーションとした。また、本ツールはVisual Studio Codeを用いて開発を行った。開発言語はJavaScript, HTML, CSSを使用した。

5.1 漢字学習AI支援ツールの設計と実装

本節では、本ツールの画面構成や学習を支援するため

の各機能の概要とその実装について述べる。

(1) 画面デザイン

開発した漢字学習AI支援ツールの画面構成を図4に示す。

基本構成は第4章で述べた通りであるが、本ツールでは画面上部に設けたツールバーによって、各機能を使用することが可能である。

(2) サインアップ・ログイン機能

学習者の進捗や履歴を個別に保存・取得するための認証基盤を提供する。サインアップ・ログイン機能の実装ではAmazon CognitoのUser Poolを利用し、ユーザーのログイン、およびトークン管理を行なった。

(3) 書字機能

画面上の漢字練習用マスに、学習者が漢字を書き込めるようにする。HTML5のcanvas要素とPointer Eventを用いて実装した。ポインタの筆記開始・移動・筆記終了に応じてイベントを取得し、各画を時系列の座標列として記録しながらリアルタイムに描画する。

学習者が書いた漢字に誤りがあった場合、容易に修正できる機能としてUndo機能とRedo機能を提供する。これにより、書き始めや書き終わりを間違えた場合や一部の線を修正したい場合において、前の状態に戻って修正したり、消去した線を再び復帰させたりしながら書き進めることが可能となる。Undo/Redo機能は、HTML5のcanvas要素とJavaScriptを用いて実装した。描画内容の履歴を管理するためにスタック構造を採用し、操作の取り消しとやり直しを実現した。

また、書き直しをする際、Undo機能により初めからやり直すと、何度も同じ操作を繰り返す必要がある。そこで一回の操作で最初から書き直しを可能とするために、全消し機能を提供する。全消し機能の実装にはCanvas APIを利用してキャンパスの内容を初期化する方法を採用した。この機能により、UndoやRedoなどのスタックやストロークデータも初期化される。

(4) 誤り箇所視覚化機能

学習者が自らの漢字の誤りを理解し修正を行えるよう支援するために、書字機能を用いて書いた漢字を生成AI

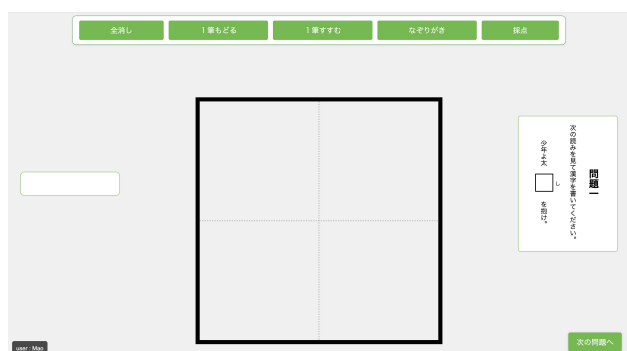


図4 漢字学習AI支援ツールの画面構成

による誤り指摘結果を基に誤り箇所を視覚化する機能を提供する。生成AIによる誤り指摘結果を受け取り、指摘に含まれる画数情報を抽出する。誤りをフィードバックエリアに指摘を表示すると共に、抽出した画数に対応するストロークをハイライトすることで誤り箇所を視覚化する（図5）。

(5) なぞり書き機能

学習者が正しい漢字を視覚的に理解し、模倣練習を行えるようにするためになぞり書きができるよう、お手本漢字を表示できる機能を提供する。

なぞり書き機能の実装にはHTML5のcanvas要素を使用し正解漢字を透かし文字として表示する仕組みを構築した。正解漢字はHTMLのdata-kanji属性から取得し、「UDデジタル教科書体NK-R」で表示される。

透かしの描画には、Canvas APIを活用し、キャンバスの大きさに合わせて示されるよう設定した（図6）。



図5 誤り箇所視覚化機能



図6 なぞり書き機能



図7 一覧表示機能

(6) 一覧表示機能

学習者がこれまでの練習結果を振り返り、自らの習熟状況を把握できるようにするために、学習履歴を蓄積・一覧表示する機能を提供する。

採点ボタンを押した時点での漢字練習用マスに書かれている漢字（画像）、フィードバックなどの各データをAWS S3に保存し、一覧形式で表示できるようにする（図7）。

6. 漢字学習AI支援ツールの評価

6.1 学習支援機能の評価

本ツールが提供する機能が漢字学習を支援するのに有用であるかどうかを検証することを目的に評価実験を行った。

6.1.1 実験概要

対象は、本学の学生14名と小学5年生1名とした。

被験者に、本ツールの各機能（Undo/Redo、なぞり書き、全消し、誤り箇所視覚化、一覧表示）を使用してもらい、図8に示す項目について、「とてもそう思う」を2点、「全くそう思わない」を-2点として、2点から-2点の5段階の点数付けによる主観的評価をしてもらった。

6.1.2 実験結果と考察

主観的5段階評価の結果を図9に示す。主観評価の結果から、本ツールに実装された各機能は漢字学習を支援するのに有用であるという肯定的な評価を得た。

自由記述欄からは、誤り箇所視覚化機能の表示方法や一覧表示機能での進捗を把握しやすくする工夫を求める改善点が得られた。

Undo/Redo機能や全消し機能、なぞり書き機能に関しては、書き間違えた際に簡単に修正できる点や書き直しを効率的に行える点、正しい漢字の字形について学べる点が高く評価された。紙と鉛筆で行うような学習にはないデジ

- Q1 Undo/Redo機能は、漢字記入時の書き誤りを修正するのに役立ちましたか。
- Q2 なぞり書き機能は、正しい漢字の形を学習するのに役立つと感じましたか。
- Q3 全消し機能は、漢字記入時の書き誤りを修正するのに役立ちましたか。
- Q4 誤り箇所は視覚的にわかりやすかったですか。
- Q5 一覧表示機能は、学習の進捗確認や復習に役立つと感じましたか。

図8 学習者を支援する機能に関する質問事項

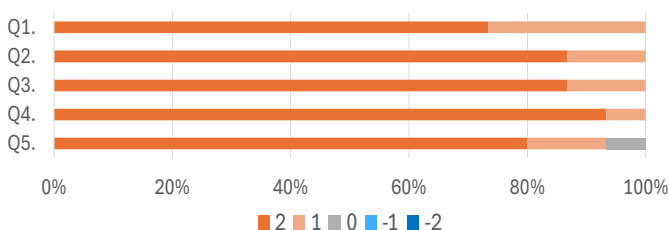


図9 学習者を支援する機能の評価結果

ルドリルの特性を活かしている部分がこのような評価の要因であると考えられる。誤り箇所視覚化機能は、誤り箇所の色を変更することで指摘箇所の視覚的なわかりやすさが表示されることについて、高い評価を得た。一方で、複数の指摘があった際に、同じ色で誤り箇所の色が変更されるため、誤り箇所の特定に困難が生じる可能性が考えられる。その他の意見としてボタンの色分けによる視認性向上や点数表示による達成感の向上などが挙げられた。このことから、デジタルドリルを用いた学習において見やすさや達成感などの学習者が得る経験を向上させることが学習を支援する際に有用であると考えられることができる。

6.2 生成AIによる指摘の評価

6.2.1 評価の目的

学習者が書いた様々な字形の漢字に対して、3章4章で設計検証した手法が字形を大きく損なう誤りを適切に判定・指摘ができていくのかについて評価を行い、本ツールが研究の目的を達成するために有用であるか否かを検証するために評価実験を行った。

6.2.2 評価方法

対象は、本学の学生14名と小学5年生1名とした。被験者に対し、本ツールにおける「正しい漢字」の定義について説明をした後に、実際に書かれた8字分の漢字とそれに対する指摘を提示した。アンケートでは、図10に示す項目について、「とてもそう思う」を2点、「全くそう思わない」を-2点として、2点から-2点の5段階の点数付による主観評価、及び選択の理由（自由記述）を回答してもらった。また、被験者が採点する立場であればどこを指摘するかについて、本研究における「正しい漢字」の定義に従って誤り箇所を記入してもらった。

6.2.3 実験結果と考察

主観評価の結果を図11に示す。

評価実験の結果から、生成AIは漢字の誤りを指摘する上で、わかりやすい文章で誤り箇所を特定する点で有用性

が示されたが、自由記述からはいくつかの課題も挙げられた。

採点する立場であればどこを指摘するかに対する回答では、本研究における「正しい漢字」の定義に従って指摘を行うよう依頼したにも関わらず、生成AIと被験者が採点する場合で指摘する箇所の相違があった。これは字形を損なう可能性のある誤りとしてどの部分が重要であるかについて生成AIが理解しきれていない可能性が考えられる。

また、指摘の内容に関して、何画目にどのような誤りがあるという指摘では、指摘されている画がおかしいといった意見もあった。たとえば、標準筆順における2、3画目を続け字で記述した場合、これらをまとめて2画目として処理してしまうため、以降の画数ラベルが標準筆順と一致なくなることが原因であったと考えられる。

また、生成AIの指摘から誤りを修正する際に、内容が曖昧であることが漢字学習において誤りを修正する際にかえって困難を生じさせる可能性があるという課題が挙げられた。これはプロンプトで誤りの指摘に関して、「わかりやすい文章」「簡潔に」というような指示を行ったことが原因であると考えられる。指摘内容において誤りがあった際にどのように修正すべきなのかについて言葉にできるよう、今後プロンプトの検証を行なっていく必要があると考えられる。

加えて、誤り箇所の視覚化についても複数の誤りがある場合に同じ色で表示されることが誤りの特定を困難にする可能性が挙げられたため、誤りの種類ごとに異なる表現を取り入れるなどの改善をする必要がある。一覧表示機能についても、全ての問題が表示されるため、どの問題を間違えたのかを把握しづらいという意見があり、誤った漢字のみを表示するような設定や、誤りの種類ごとに分類する機能を追加することで、振り返り学習の効果を高めることができると考えられる。

7. おわりに

本研究では、字形を大きく損なう誤りの判定・指摘を行う漢字学習ツールを開発することを目標として、生成AIに少量の入力データと正解データ、プロンプトを渡すことで漢字の誤りを判定・指摘する手法を提案し、その実装と評価を通して、提案した手法が学習者の漢字学習を支援するのに有用であるかを検証した。

提案した誤り判定手法は、従来のパターンマッチングや機械学習を用いた手法とは異なり、学習者が書いた手書き漢字と正解漢字を表す少量のデータを生成AIに渡すだけで、字形を大きく損なう誤りを判定・指摘できる可能性が示唆された。

提案した誤り判定手法を実装した漢字学習AI支援ツールの評価実験の結果、指摘の内容と誤り箇所の特定に関して有用であることが示された一方、指摘の精度や視覚化方

Q1 誤り指摘の内容は適切だと感じましたか

Q2 指摘の内容は、学習者にとってわかりやすい文章でしたか。

Q3 指摘の文章内で誤り箇所は特定されていますか。

図10 生成AIの指摘に関する質問事項

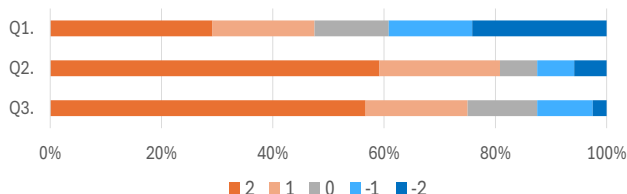


図11 生成AIによる指摘の評価結果

法について検討が必要であることが示された。

今後は、評価実験の課題として挙げた生成AIによる誤り指摘内容の曖昧さの改善や精度の向上を目指す。特に生成AIが誤りを指摘する際の一貫性を確保し、学習者がどのように修正すれば良いのかを明確に伝えられるよう、プロンプトの設計を見直す必要がある。また、指摘方法に関しても学習者の書いた漢字を基準とするか、正しい漢字を基準とするかについて検討を行い、より適切なフィードバックを提供できる仕組みを構築することが求められる。

また、使用する生成AIについても精度向上のためにファインチューニングや、データの読み取りに長けたものを選定する必要がある。さらに、教材として使用されることを考え、費用が抑えられるものについても十分検討する必要がある。

参考文献

- [1] 文部科学省：義務教育段階における1人1台端末の整備状況（令和4年度末時点）
https://www.mext.go.jp/content/20230711-mxt_shuukyo01-000009827_01.pdf
- [2] 文部科学省：GIGAスクール構想のもとの校務DX化チェックリスト https://www.mext.go.jp/content/20240402-mxt_jogai01-000033278_01.pdf
- [3] 文化庁：常用漢字表の字体・字形に関する指針（報告）
https://www.bunka.go.jp/seisaku/bunkashingikai/kokugo/hokoku/pdf/jitai_jikei_shishin.pdf
- [4] Wes Gurnee, Max Tegmark: LANGUAGE MODELS PRPRESENT SPACE AND TIME, arXiv preprint arXiv:2310.02207v3 [cs.LG] (2024)
- [5] 古積拓見, 稲谷壮一郎, 蔡文杰, 中川正樹: 漢字を正しく綺麗に書くことを学ぶための学習システム, 情報処理学会研究報告, 情報処理学会研究報告, Vol.2014,No.14,pp15-22 (2014)
- [6] 堀田龍也, 中沢美仁, 長勢美里, 山口直人, 高橋純: 小学生の漢字書き取りにおける誤答の分布及び教員の採点における重要度に関する分析, 日本教育工学会研究報告集, JSET15-3, pp175-182 (2015)